Design Document

Table of Contents

Design Document

Overview

The objective of this assignment is to implement and test a stand-alone Linux firewall and packet filter. The firewall should be created using Netfilter and must follow a basic set of rules. This project will include a design and testing document which proves the functionality of the software. The standalone firewall will allow users to hide and protect servers from external networks and allow users to define which ports or types of ports to open and close for TCP, UDP and ICMP.

Firewall Design

The firewall has been designed with set of default policies to DROP all packets that come through on the INPUT, OUTPUT and FORWARD chains except for the following:

- Inbound/Outbound TCP packets on allowed ports
- Inbound/Outbound UDP packets on allowed ports
- Inbound/Outbound ICMP packets based on type numbers
- All TCP connections that belong to an existing connection (on allowed ports)
- Fragments

All other traffic that comes through the firewall will be dropped by default including the following:

- All packets that fall through the default rule
- All packets destined for the firewall host from the outside
- Any packets with a source address from the outside matching your internal network
- All inbound traffic that is coming the “wrong” way
- All TCP packets with the SYN and FIN bit set
- Telnet packets

Design Document

- All external traffic directed to ports 32768 – 32775, 137 – 139 and TCP ports 111 and 515

Project Constraints

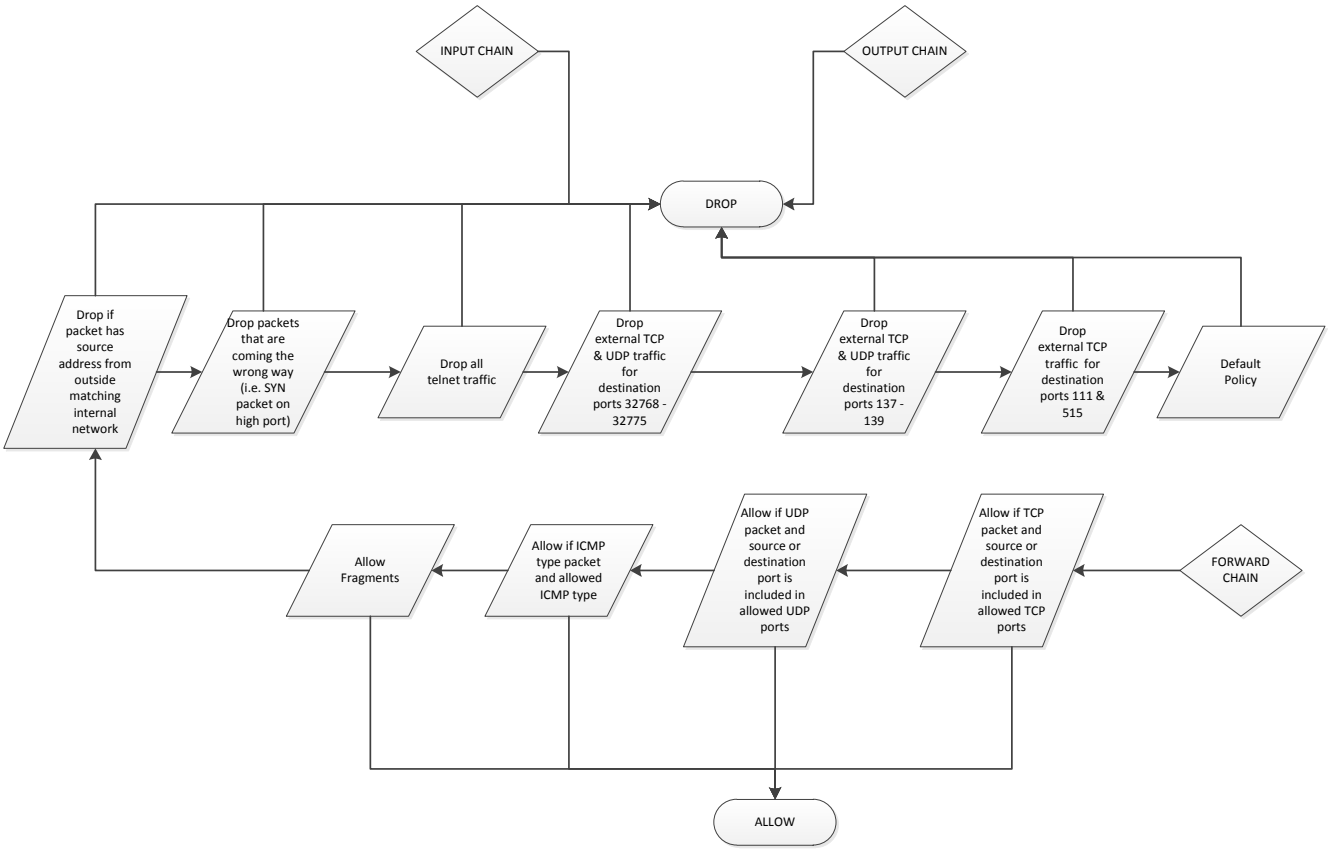
- The firewall/packet filter must be designed and implemented using Netfilter.
- The firewall script must have two sections: a "User Configurable Section" and an "Implementation Section".
- The user configuration section will allow a user to set at least the following parameters:

    - Name and location of the utility you are using to implement the firewall.
    - Internal network address space and the network device.
    - Outside address space and the network device.
    - TCP services that will be allowed.
    - UDP services that will be allowed.
    - ICMP services that will be allowed.
    - Allowing only NEW and ESTABLISHED traffic to go through the firewall.

- The firewall must reject those connections that are coming the "wrong" way, inbound connection requests (unless of course it is to a permitted service).
- The firewall must include test scripts designed to validate the firewall rules.

Design Document

Design



Design Document

Pseudo Code

If incoming packet on INPUT chain
    DROP

If outgoing packet on OUTPUT chain
    DROP

If Forwarded TCP packet && sport || dport is $allowedTcpPorts
    ACCEPT

If Forwarded UDP packet && sport || dport is $allowedUdpPorts
    ACCEPT

If Forwarded ICMP packet == $allowedIcmpTypes
    ACCEPT

If Forwarded packet is fragments
    ACCEPT

If forwarded packet $sourceAddress == $internalNetworkAddress
    DROP

If forwarded packet is coming the wrong way
    DROP

If forwarded packet sport || dport == port 22
    DROP

If forwarded packet type is TCP || UDP && dport is between 32768 – 32775
    DROP

If forwarded packet type is TCP || UDP && dport is between 137 – 139

Design Document

DROP

If forwarded packet type is TCP && dport is 111 || 515
DROP

If forwarded packet != firewall rule
DROP

If dport == 21 || 22
Set TOS to minimize-delay

If dport == 20
Set TOS to maximize-throughput

Jivanjot S. Brar & Shan Bains A00774427 & A00737179

Design Document

Project Files

Assign2-Network-Setup.sh
Firewall_Rules.sh
Monitor.sh
Test_Script.sh
Assign2-Design-Document.pdf
Assign2-Testing-Document.pdf

How to setup the gateway on a client machine

Run Assign2-Network-Setup.sh
Choose the workstation option (option 1)

How to setup the standalone firewall

Run Assign2-Network-Setup.sh
Choose the firewall option (option 2)
Choose yes to deploy the firewall rules (option 1)

How to run testing script

Run Test_Script.sh:

Usage: chmod +x Test_Script.sh
./Test_Script.sh > Test_Results.txt
View Test_Results.txt